Name: *Student id:* *Section:* *Serial#:*

Question #	1	2	3	4	Total
Max points	15	10	18	20	63
Points earned					

University of Bahrain *College of Information Technology*
Department of Computer Science

ITCS242: Assembly Language Programming

Second test *Date: JAN 04, 2016* *Time: 90 minutes*
****

QUESTION ONE: Answer each of the following questions as stated.

1) Give no more than **6** instructions to store in **edx** the product of multiplying the top two words stored on the stack. **{5 pts}**

```
pop   ax
pop   dx
imul  dx
shl   eax, 16
shld  edx, eax, 16
```

2) Give no more than **5** instructions to calculate: *ecx* = left half of eax + left half of ebx. **{5 pts}**

```
sar   eax, 16
sar   ebx, 16
add   eax, ebx
mov   ecx, eax
```

3) Write no more than **9** instructions to store in ***AX*** the count of semicolons ";" in a string ***strX***.
strX byte "ab7;89,G;b9,56:tr,Y#Q,;,90;: F... " . **{5 pts}**

```
     mov   ax, 0
     mov   ecx, sizeof strx
     mov   ebx, 0
L1:  cmp   strx[ebx], ";"
     jne   L2
     inc   ax
L2:  inc   ebx
     loop  L1
```

Name: *Student id:* *Section:* *Serial#:*

QUESTION TWO: What will be in the indicated registers after executing each of the following codes?
{10*1 pts}

a)
```
MOV  CX,7A04H
MOV  BX,9A6CH
SAR  BX, CL
```
BX = **F9 A6** H

b)
```
MOV    AL, 20H
MOV    BL, 96H
IMUL   BL
```
AX = **F2 C0** H

c)
```
MOV    AX, 7A9EH
XOR    AX, 0F5F5H
```
AX = **8F 6B** H

d)
```
MOV    BX, 9C5FH
MOV    AX, 7B6AH
SHLD   AX, BX, 8
```
AX = **6A 9C** H

e)
```
MOV  AX,37A9H
MOV  BX,5F4DH
TEST AX, BX
ROR  BX, 4
```
BX = **D5 F4** H

f)
```
MOV   SP, 1A5CH
POP   BX
POP   ECX
```
SP = **1A 62** H

g)
```
MOV    AX,0EA5CH
MOVSX  AX, AL
MOV    BX, AX
SAL    AX, 4
ADD    AX, BX
```
AX = **O6 1C** H

h)
```
XOR   AX,AX
JZ    L1
SUB   AX,0EEEEH
JMP   L9
L1:ADD  AX,0AC7FH
L9:
```
AX = **AC 7F** H

i)
```
MOV  AX,0A386H
MOV  BX,0FFFFH
CWD
IDIV BX
```
AX = **5C 7A** H

j)
```
MOV   AX,4AFFH
MOV   BX,7C5EH
CMP   AL, BL
JL    L3
IMUL  BH
JMP   L4
L3:NEG  BH
L4:INC  BL
```
BX = **84 5F** H

Name: *Student id:* *Section:* *Serial#:*

QUESTION THREE: Convert each of the following C++ codes into Assembly language.

1) *signed byte x, y;* ***// You have to properly define f only*** **{6 pts}**
short f;
f = (x % y) ++;

```
f       sbyte     ?
        movsx     ax, x
        idiv      y
        inc       ah
        mov       f, ah
```

2) *int j=0, ct = 0, x[64]= {24,-17,...};* **{6 pts}**
while (j < 48) ***// Assume that x and ct are already defined:***
{ if (x[j] >= 0) ***// x sdword 10, -12, ...***
ct++; ***// ct sdword 0***
j++;
}

```
        mov       ebx, 0              ; index
wh:     cmp       ebx, lengthof x
        jGE       fin
        cmp       x[4*ebx], 0
        jL        L2
        inc       ct
L2:     inc       ebx
        jmp       wh
fin:
```

3) *void funU (byte *a, byte *b)* **{6 pts}**
*{ byte t = *a;*
**a = *b;*
**b = t;*
}

```
funU    proc      uses esi edi ax, a: ptr byte, b: ptr byte
        mov       esi, a
        mov       edi, b
        mov       ah, [esi]
        xchg      ah, [edi]
        mov       [esi], ah
        ret
funU    endp
```

Name: *Student id:* *Section:* *Serial#:*

QUESTION FOUR: **{20 pts}**

Write a complete Assembly program that defines in the data segment a variable ***digits*** and a string ***strMe*** consisting of up to 100 characters. The program consists of the following two procedures:

a) The procedure ***funx*** accepts 2 parameters: a pointer to string ***strx***, and its length ***y***, calculates and returns the count of Arabic digits (*f*) in the string ***strx***. *(Write the procedure FUNX in a form that allows using invoke statement).*

b) The procedure **main** prompts the user to read from the KBD a string ***strMe***, at the beginning of a new line, displays all characters in ***strMe*** separated by a space, calls the procedure **FUNX** to store in a variable ***digits*** the count of Arabic digits in ***strMe***.

```
          INCLUDE   IRVINE32.INC
          .data
strMe     byte      101 DUP(?)
digits    word      ?
strU      byte      "Enter a string of upto 100 characters: ",0
          .code
;    ********************************************
FUNX      PROC      USES ESI EDI ECX, STRX:PTR BYTE, Y:DWORD, F:PTR WORD
          MOV       ESI, STRX
          MOV       EDI, F
          MOV       ECX, Y
          MOV       WORD PTR [EDI],0
     L1:  CMP       byte ptr[ESI],'0'
          JB        L2
          CMP       byte ptr[ESI],'9'
          JA        L2
          INC       WORD PTR [EDI]
     L2:  INC       ESI
          LOOP      L1
          RET
FUNX      ENDP
;    ********************************************
MAIN      PROC
          LEA       EBX, STRU
          CALL      WRITESTRING
          LEA       EDX, STRME
          MOV       ECX,101
          CALL      READSTRING
          MOV       EDX, EAX
          MOV       ECX, EAX
          MOV       EBX, 0
          CALL      CRLF
   LU:    MOV       AL, STRME[EBX]
          CALL      WRITECHAR
          MOV       AL, 20H
          CALL      WRITECHAR
          INC       EBX
          LOOP      LU
          INVOKE    FUNX, ADDR STRME, EDX, ADDR DIGITS
          EXIT
MAIN      ENDP
          END       MAIN
```